UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/693,834	10/24/2003	Trishul Chilimbi	3382-66135-01	1003

26119 7590 07/25/2007

KLARQUIST SPARKMAN LLP
121 S.W. SALMON STREET
SUITE 1600
PORTLAND, OR 97204

EXAMINER	
WEI, ZHENG	

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
07/25/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/693,834	CHILIMBI ET AL.	
	Examiner	Art Unit	
	Zheng Wei	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 May 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-25* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *24 October 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *03/07/2007, 05/11/2007*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Remarks

1. This office action is in response to the amendment filed on 05/11/2007.

2. Claims 1, 3, 6, 7 and 12 have been amended.

3. The objection to claims 1 and 6 is withdrawn in view of the Applicant's amendment;

4. The 35 U.S.C. 112 second paragraph rejection of claims 1-5, 7-10 and 12-13 is withdrawn in view of the Applicant's amendment.

5. Claims 1-25 remain pending and have been examined.

Information Disclosure Statement

6. The information disclosure statements filed on 03/07/2007 and 05/11/2007 have been reviewed and placed in the application file.

Response to Arguments

7. Applicant's arguments filed on05/11/2007, in particular on pages 8-17, has been fully considered but they are not persuasive. For example:

   - At pages 8-10, section Double Patenting Rejection, Applicant contends that the office action fails to make a establish proper case of nonstatutory obviousness-type double patenting over claim 1 of Chilimbi. Because Chilimbi does not recite "sampling rates" and "frequency". However, the Examiner respectfully disagrees with that. As to previous Office action, paper 5, lines 1-

2, the Examiner pointed out "tracking a number of iterations of the check code executed" in Chilimbi, does anticipate "tracking a frequency of execution" in applicant's claim 1. Because the "tracking frequency" can be reasonable interpreted as "a number of iterations" which means times of execution. The Examiner also pointed out and highlighted at page 5, "switching between checking and profiling phases upon the tracked number of iteration" in Chilimbi, does indicate that the execution times/sampling rate between checking and profiling phases has to be changed based on the number of iteration/frequency. Therefore, the Examiner reasserted that the obviousness-type double patenting to claim 1 over claim 1 of Chilimbi is proper. For the same reason the double patenting rejection to claim 6, 18 and 23 is also maintained as set forth in the previous Office Action.

- At page 11, section Rejections Under 35 U.S.C. 112, Applicant does not respond to the rejection of claim 11 about relative term "last access". Therefore, the 112 rejection to claim 11 is maintained. The rejection to claims 1-5 and 7-10 and 12-13 are withdrawn in view of the Applicant's amendment.
- At page 12, last paragraph, the Applicant points out that Wu does not teach or suggest "adapting the sampling rate" as recited in claim 1. However, the Examiner respectfully disagrees. As to previous Office action, paper number 11, the Examiner pointed out at Fig.5A, Fig.5B and related text at col.9, lines 1-20 clearly discloses: "The edge has been executed at least HOT_THRESH OLD times... In other words, when counter(e) overflows, the edge e that is

identified as cold in the previous profile phase (sample) becomes hot in the current phase... processing block 540 where new phase transition information is generated. At processing block 545, the dynamic optimizer may take over and use the new edge profile information (new sample) to re-optimize the program (adapting). Flow then returns to start block 501" for execution (emphasis added). Therefore, the trigger counter(e) does cause re-optimization of code to change edge(e)'s execution time and further teach adapting the sampling rate/execution times as recited in claim 1. Thus, Wu does anticipated claimed limitation as set forth in the previous Office Action.

- At page 13, section about claim 3, Applicant contends that Wu does not teach or suggest "subsequently, selectively reducing a frequency at which the duplicated version is executed". However, Wu does disclose, as the Examiner cited at office action page number 12 about "Decrement Trigger Counter" and related text at Wu col.9, lines 1-20, "the trigger_counter is decremented at processing block 525" (reducing) when "the edge e has been executed at least HOT_THRESH OLD times" (duplicating). Therefore, Wu does anticipate the all the limitation of claim 3.
- At page 13, section about claim 6, Applicant argues that Wu dose not disclose "the frequency at which the burst are performed decreases as the number of execution s of either the original procedure or copy of the procedure is executed". However, as the Applicant cited in previous Office Action about Wu's Fig.2 and related text at col.5, lines 10-33, Wu discloses that "at block

240, the dynamic optimizer optimized the 'hot ' code and at block 220, re-execute program; Moreover, at Fig.5 and related text at col.9, lines 1-20, Wu also discloses "Decrement Trigger Counter". Therefore, the rejection to claims 6 is maintained.

- At page 14, section about claim 14, Applicant points out that Wu does not disclose each and every element of claim 14 about additional program execution. However, Wu at Figl.5B also discloses "Increment Counter" and "Decrement Counter" which are used to increase/decrease counter according the execution times, then Trigger Counter as signal phase transition and re-optimize program for re-execution. Therefore, Wu does disclose all the limitation about claim 14.
- At page 15-16, sections about claims 7, 18 and 23. For the same reason as address above, refers to Fig.5A and Fig5b and related text about "Increment/Decrement Counter, Trigger Counter and Re-Optimize program, Wu does discloses all the limitation about frequently sampling issues in claims 7, 18 and 23. Therefore, the rejection to claims 7, 18 and 23 are maintained.

Claim Rejections - 35 USC § 112

8. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

9. Claim 11 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 11: The term '"last access"' in claim is a relative term which renders the claim indefinite. The term '"last access"' is not defined by the claim, the specification does not provide a standard for ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention. For the purpose of compact prosecution, the Examiner treats '"last access"' as –any instrumenting information--

10. Claims 1, 6, 18 and 23 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claim 1 of U.S. Patent No. 7,140,008. Although the conflicting claims are not identical, they are not patentably distinct from each other. As can be seen from the table below, instant claims and the claims of U.S. Patent are directed to the same subject matter of the invention. For example,

Instant Application 10/693834	**U.S. Patent** 7,140,008
Claim 1. **A method of instrumenting a program** to provide instrumentation data, the method comprising:	Claim 1. **A method of instrumenting a program** to provide sampled temporal profiling bursts of a program execution trace, the method comprising:

creating an instrumented version of the program comprising duplicate versions of at least some code paths in the programs, such that a duplicate code path has an original version code path and an instrumented version code path with instrumentation code for capturing instrumentation data;	**providing a duplicate version of at least some procedures in the program** with instrumentation for capturing a temporal sequence of data references by the program;
	inserting check code at locations of at least some procedure entries and loop back-edges of the program;
tracking a relative frequency of execution of the code paths;	alternately **tracking a number of iterations** of the check code executed in a checking phase and a profiling phase up to respective checking and profiling count parameters, wherein the profiling count parameter is more than one and the duplicate version of at least some procedures with instrumentation are executed during the profiling phase and a non-instrumented version of the program's procedures are executed during the checking phase;
when a code path is to be executed, **determining to** dispatch execution into the instrumented version code path at a sampling rate for the respective code path and otherwise into the original version code path;	upon executing the check code when in **the checking phase**, causing execution to proceed in the non-instrumented version of the program's procedures; upon executing the check code when in the profiling phase, causing execution to proceed in the duplicate instrumented version of the at least some procedures; and

adapting the sampling rate for the code paths according to the relative frequency of execution of the code paths.	**switching** between checking and profiling phases upon the tracked **number of iterations** of the check code reaching the respective count parameter of the respective phase.
Claim 6. **A method of instrumenting a computer program** containing procedures, the method comprising: **creating a copy of at least some of the original procedures** in the computer program; **inserting instrumentation** into the copies; creating an executable version of the program containing the original procedures and the copies;	Claim 1. **A method of instrumenting a program** to provide sampled temporal profiling bursts of a program execution trace, the method comprising: **providing a duplicate version of at least some procedures in the program** with instrumentation for capturing a temporal sequence of data references by the program; **inserting check code** at locations of at least some procedure entries and loop back-edges of the program; alternately tracking a number of iterations of the check code executed in a checking phase and a profiling phase up to respective checking and profiling count parameters, wherein the profiling count parameter is more than one and the duplicate version of at least some procedures with instrumentation are executed during the profiling phase and a non-instrumented version of the program's procedures are executed during the

	checking phase;
executing the executable version of the program, wherein the copies of the procedures are executed in bursts, and the frequency at which the bursts are performed decreases as the number of executions of **either the original procedure or copy of the procedure** is executed.	upon **executing the check code** when in the checking phase, causing execution to proceed in the **non-instrumented version** of the program's procedures; upon executing the check code when in the profiling phase, causing execution to proceed in **the duplicate instrumented version** of the at least some procedures; and switching between checking and profiling phases upon the tracked number of iterations of the check code reaching the respective count parameter of the respective phase.
Claims 18, 23. **A method of instrumenting software**, the method comprising: **producing a copy of at least some procedures of the software**; **inserting instrumentation** into the copies; and	Claim 1. **A method of instrumenting a program** to provide sampled temporal profiling bursts of a program execution trace, the method comprising: **providing a duplicate version of at least some procedures in the program** with instrumentation for capturing a temporal sequence of data references by the program; **inserting check code** at locations of at least some procedure entries and loop back-edges of the program;

sampling a copy of a procedure at a rate inversely proportional to how frequently the procedure is executed/**sampling a copy of a procedure** at higher rates for procedures executed less frequently and sampling a copy of a procedure at lower rates for procedures executed more frequently.	alternately **tracking a number of iterations of the check code executed in a checking phase** and a profiling phase up to respective checking and profiling count parameters, wherein the profiling count parameter is more than one and the duplicate version of at least some procedures with instrumentation are executed during the profiling phase and a non-instrumented version of the program's procedures are executed during the checking phase; upon executing the check code when in the checking phase, causing execution to proceed in the non-instrumented version of the program's procedures; upon executing the check code when in the profiling phase, causing execution to proceed in the duplicate instrumented version of the at least some procedures; and switching between checking and profiling phases upon the tracked number of iterations of the check code reaching the respective count parameter of the respective phase.

Claim Rejections - 35 USC § 102

11. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

12. Claims 1-6, 14, 18, 19 and 23 are rejected under 35 U.S.C. 102(e) as being anticipated by Wu (Youfeng Wu, US 7,032,217 B2).

Claim 1:

Wu discloses a method of instrumenting a program to provide instrumentation data, the method comprising:

- creating an instrumented version of the program comprising duplicate versions of at least some code paths in the programs, such that a duplicate code path has an original version code path and an instrumented version code path with instrumentation code for capturing instrumentation data (see for example, Fig.2, step 210 and related text, also see col.4, line 64-col.5, line 20);
- tracking a relative frequency of execution of the code paths (see for example, Fig.5A, Fig.5B, steps 505-530, steps 555-585 and related text about "profile Counter");
- when a code path is to be executed, determining to dispatch execution into the instrumented version code path at a sampling rate for the respective code path and otherwise into the original version code path (see for example, Fig.5A, Fig.5B, steps 530, 585 and related text); and

- adapting the sampling rate for the code paths according to the relative frequency of execution of the code paths (see for example, Fig.5A, Fig.5B, steps 530, 585 and related text about "Trigger Counter").

Claim 2:

Wu further discloses the method of claim 1 wherein instrumentation data comprises data relating to runtime data references, branch executions, memory allocations, synchronization events, data loads, data stores, or branches (see for example, Fig.3, element 320 and related text, also see p.3, line1 "three branch instructions").

Claim 3:

Wu discloses a method of instrumenting a program to provide runtime program data, the method comprising:

- providing a duplicate version of at least some already present procedures in the program with instrumentation for capturing runtime program data (see for example, Fig.2, step 210 and related text, also see col.4, line 64-col.5, line 20);
- executing the duplicate version of at least some of the procedures (see for example, Fig.2, step 220, "Program execution" and related text); and
- subsequently, selectively reducing the frequency at which the duplicate version is executed (see for example, Fig.5A, step 525 "Decrement Trigger

Counter" and related text).

Claim 4:

Wu further discloses the method of claim 3 wherein the frequency at which the duplicate version is executed is reduced at a rate inversely proportional to how frequently a procedure of the software is executed (see for example, Fig.5B, steps 570, 575, 576 and 580 "Decrement/Increment Counter" and related text).

Claim 5:

Wu also discloses the method of claim 3 wherein the frequency at which the duplicate version is executed is reduced as a function of how frequently a procedure of the software is executed (see for example, Fig.2, step 240, 250 and related text, also see Fig.5A, steps 54, 545, "Generate New Phase Transition Information" and related text).

Claim 6:

Wu discloses a method of instrumenting a computer program containing procedures, the method comprising:

- creating a copy of at least some of the original procedures in the computer program (see for example, Fig.2, step 210 and related text, also see col.4, line 64-col.5, line 20);

- inserting instrumentation into the copies (see for example, Fig.2, step 210 and related text, also see col.4, line 64-col.5, line 20);
- creating an executable version of the program containing the original procedures and the copies (see for example, Fig.2, steps 210-220 and related text, also see col.4, line 64-col.5, line 20 about "compiler");
- executing the executable version of the program, wherein the copies of the procedures are executed in bursts, and the frequency at which the bursts are performed decreases as the number of executions of either the original procedure or copy of the procedure is executed (see for example, Fig.2, step 210 and related text, also see col.4, line 64-col.5, line 20, also see Fig.5A, step 525, "Decrement Trigger Counter" and related text).

Claim 14:

Wu discloses a method of analyzing software, the method comprising:

- creating an instrumented version of the software containing an original version and an instrumented version of at least some procedures in the software, wherein the instrumented versions comprise instrumentation points (see for example, Fig.2, step 210 and related text, also see col.4, line 64-col.5, line 20);
- inserting additional programming code at the instrumentation points that produce runtime information when executed (see for example, Fig.2, step 210 and related text, also see col.4, line 64-col.5, line 20); and

- executing the instrumented version of the software, wherein the additional programming code is executed more frequently when located at instrumentation points that are less frequently executed, and the additional programming code is executed less frequently when located at instrumentation points that are more frequently executed (see for example, Fig.2, step 210 and related text, also see col.4, line 64-col.5, line 20, also see Fig.5A, step 525, "Decrement Trigger Counter" and related text).

Claim 18:

Wu discloses a method of instrumenting software, the method comprising:

- producing a copy of at least some procedures of the software (see for example, Fig.2, step 210 and related text, also see col.4, line 64-col.5, line 20);
- inserting instrumentation into the copies (see for example, Fig.2, step 210 and related text, also see col.4, line 64-col.5, line 20); and
- sampling a copy of a procedure at a rate inversely proportional to how frequently the procedure is executed (see for example, Fig.5A, Fig.5B, steps 530, 585 and related text about "Trigger Counter").

Claim 19:

Wu further disclose the method of claim 18, wherein the instrumentation stores data relating to the software when executed (see for example, col.5, lines 40-41,

"store the counter back to memory").

Claim 23:

Wu discloses a method of instrumenting software, the method comprising:

- producing a copy of at least some procedures of the software (see for example, Fig.2, step 210 and related text, also see col.4, line 64-col.5, line 20);
- inserting instrumentation into the copies (see for example, Fig.2, step 210 and related text, also see col.4, line 64-col.5, line 20); and
- sampling a copy of a procedure at higher rates for procedures executed less frequently and sampling a copy of a procedure at lower rates for procedures executed more frequently (see for example, Fig.5A, Fig.5B, steps 530, 585 and related text about "Trigger Counter").

Claim Rejections - 35 USC § 103

13. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

14. Claims 7-13 and 15-17are rejected under 35 U.S.C. 103(a) as being unpatentable over Wu (Youfeng Wu, US 7,032,217 B2) in view of Alexander (Alexander et al., US 6,658,652 B1).

Claim 7:

Wu discloses a method for detecting memory leaks in software, the method comprising:

- creating an instrumented version of the software containing an original version and an instrumented version of each procedure in the software (see for example, Fig.2, step 210 and related text, also see col.4, line 64-col.5, line 20);
- executing the instrumented version of the software, wherein the instrumented version of the procedures are sampled at higher rates for procedures executed less frequently and sampled at lower rates for procedures executed more frequently (see for example, Fig.2, step 210 and related text, also see col.4, line 64-col.5, line 20, also see Fig.5A, step 525, "Decrement Trigger Counter" and related text);
- storing instrumentation data obtained by execution of the instrumented version of the software (see for example, Fig.4, element 460, "Profile operations buffer" and related text);

But does not disclose reporting all objects that satisfy a staleness predicate as memory leaks. However, Alexander in the same analogous art of program tracing using shadow heap memory leak detection and other heap analysis

discloses (see for example, Fig.30, JVM HEAP 3002, objects 3062-3068 and related text). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use Wu's method to further implement memory leaks detection function to test memory leaks. One would have been motivated to do so as to provide a method for accurate memory leak detection in an object-oriented environment during real-time trace processing as Alexander suggested at col.3, lines 33-37.

Claim 8:

Wu and Alexander disclose the method of claim 7, Alexander further discloses, wherein instrumentation data comprises heap allocation, heap free and heap access information (see for example, Fig.3B, element 372 "Heap" and related text, also see Fig.31, step 3114 "Profiler finds the proper slot in the shadow heap based on the relative position of the corresponding object in the heap" and related text).

Claim 9:

Wu and Alexander disclose the method of claim 7, Alexander further discloses, wherein reporting all objects comprises reporting the heap object, responsible allocation, heap frees that deallocated objects created at that allocation site, and the last access to the leaked object (see for example, Fig.32, steps 3202-3216,

"Object deallocation" and related text).

Claim 10:

Wu and Alexander disclose the method of claim 9, Alexander further discloses the method of claim 9, generating report for heap objects including the information of the last access to a leaked object (see for example, Fig.34, Fig.35, example reports and related text).

Claim 11:

Wu and Alexander disclose the method of claim 7, Alexander further discloses the method comprising creating mapping information from the software to facilitate "last access" information (see for example, Fig.28 about data structure to facilitate tracking additional information related to a routine using heap and related text".

Claim 12:

Wu and Alexander disclose the method of claim 7, Alexander further discloses, wherein the staleness predicate comprises determining whether an object on the heap has not been accessed within a predetermined length of time (see for example, Fig.10c-10D and related text).

Claim 13:

Wu and Alexander disclose the method of claim 7, Wu further discloses, wherein the instrumented version of the procedures are sampled at a rate inversely proportional to how frequently a procedure is executed (see for example, Fig.5B, steps 570, 575, 576 and 580 "Decrement/Increment Counter" and related text).

Claims 15 and 17:

Wu discloses the method of claim 14, but does not discloses, wherein runtime information comprises data relating to memory leaks or invariance. However, Alexander in the same analogous art of program tracing using shadow heap memory leak detection and other heap analysis discloses (see for example, Fig.30, JVM HEAP 3002, objects 3062-3068 and related text). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use Wu's method to further implement memory leaks detection function to test memory leaks. One would have been motivated to do so as to provide a method for accurate memory leak detection in an object-oriented environment during real-time trace processing as Alexander suggested at col.3, lines 33-37.

Claim 16:

Wu discloses the method of claim 14, but does not discloses, wherein runtime information comprises data relating to data races. However, Alexander in the same analogous art of tracing software program discloses runtime information

comprises data relating to data races (see for example, col.16, lines 27-36, "routine B" and its status: interrupt or suspended or blocked... and related text). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to further data race information that can be used to prevent data race.

15. Claims 20-22 and 24-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wu (Youfeng Wu, US 7,032,217 B2) in view of Zorn (Zorn et al., "A Memory Allocation Profiler for C and Lisp Programs")

Claim 20:

Wu discloses the method of claim 19, but does not disclose the method further comprising providing the stored data to a tool for analysis. However, Zorn in the same analogous art of error detecting discloses a tool "mprof" which is used to study the memory allocation behavior of programs. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to pass data to "mprof" for the purpose of monitor. One would have been motivated to do so to monitor executing programs and display to user as suggested by Zorn (p.1, Introduction, "allows programmer to identify where and why memory is being allocated in a program.")

Claim 21:

Wu and Zorn disclose the method of claim 20, Zorn further discloses, wherein the tool detects memory leaks (see for example, p.3, section 2 Using mprof).

Claim 22:

Wu and Zorn disclose the method of claim 20, however, neither of them explicitly discloses the tool detects data races. However, it is well known in the computer art that the operating system which the tool is running provide the same functionality about data race detection and /or protection. Therefore, this claim is obvious by Wu and Zorn.

Claim 24:

Wu discloses the method of claim 23, but does not disclose the method further comprising providing the data to software to a tool. However, Zorn in the same analogous art of error detecting discloses a tool "mprof" which is used to study the memory allocation behavior of programs. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to pass data to "mprof" for the purpose of monitor. One would have been motivated to do so to monitor executing programs and display to user as suggested by Zorn (p.1, Introduction, "allows programmer to identify where and why memory is being allocated in a program.")

Claim 25:

Wu and Zorn disclose the method of claim 24, Zorn further discloses, wherein the tool uses the communicated data to analyze the software (see for example, p.3, section 2 Using mprof).

Conclusion

16. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

17. Applicant's arguments with respect to claims rejection have been considered but are not persuasive. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a). A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

18. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Zheng Wei whose telephone number is (571)

270-1059 and Fax number is (571) 270-02059. The examiner can normally be reached on Monday-Thursday 8:00-15:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571- 272-1000.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ZW

TUAN DAM
SUPERVISORY PATENT EXAMINER